Improving Entity Recommendation with Search Log and Multi-Task Learning

Jizhou Huang[†,‡*], **Wei Zhang**[‡], **Yaming Sun**[‡], **Haifeng Wang**[‡], **Ting Liu**[†]
[†]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China
[‡]Baidu Inc., Beijing, China
{huangjizhou01, zhangwei32, sunyaming, wanghaifeng}@baidu.com, tliu@ir.hit.edu.cn

Abstract

Entity recommendation, providing search users with an improved experience by assisting them in finding related entities for a given query, has become an indispensable feature of today's Web search engine. Existing studies typically only consider the query issued at the current time step while ignoring the in-session preceding queries. Thus, they typically fail to handle the ambiguous queries such as "apple" because the model could not understand which apple (company or fruit) is talked about. In this work, we believe that the in-session contexts convey valuable evidences that could facilitate the semantic modeling of queries, and take that into consideration for entity recommendation. Furthermore, in order to better model the semantics of queries, we learn the model in a multi-task learning setting where the query representation is shared across entity recommendation and context-aware ranking. We evaluate our approach using large-scale, real-world search logs of a widely used commercial Web search engine. The experimental results show that incorporating context information significantly improves entity recommendation, and learning the model in a multi-task learning setting could bring further improvements.

1 Introduction

Over the past few years, major commercial Web search engines have enriched and improved user experience of information retrieval by proactively presenting related entity recommendations for a query along with the regular Web search results. Figure 1 shows an example of Baidu[1] Web search engine's entity recommendation results for the query "Chicago" presented on the right panel of its Web search result page.[2]

Existing studies [Blanco *et al.*, 2013; Yu *et al.*, 2014; Bi *et al.*, 2015; Huang *et al.*, 2018] in entity recommendation typically consider the query being issued at each time step independently, while ignoring the in-session context queries. A main common drawback of these approaches is that they cannot handle well the ambiguous queries, because they do not have informative evidences other than the query itself for disambiguating the meaning of entities with the same surface form. Therefore, existing entity recommendation systems tend to recommend entities with regard to the frequently asked meaning. We believe that in-session preceding context queries are valuable evidences to tackle this problem. First, the contexts convey additional insights into a user's current information need, which enables us to provide the user with more relevant entity recommendations for ambiguous queries. For example, a user's search intent behind the query "Chicago" could be either a city, a movie or a rock band. Without context information, the entities recommended for this query are mainly based on its most frequent meaning, as shown in Figure 1. However, if a query "Dreamgirls" is submitted before "Chicago" by the user, it is very likely that the user is interested in the movie rather than the city. Second, our empirical study on a sample of search logs reveals that contexts can also be beneficial for adapting recommendations to individual search needs, thus providing personalized entity recommendations. For example, if the preceding queries issued by a user before "James Cameron" are movies such as "Avatar" and "Titanic", it is observed from the subsequent search behaviors that she is more interested in the movies related to "James Cameron". In comparison, if the preceding queries are celebrities, she is more interested in the celebrities related to "James Cameron".

Recently, in order to take into account the in-session con-

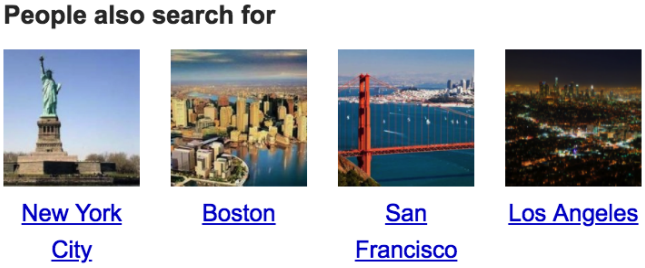

Figure 1: Example of Baidu's entity recommendation results for the query "Chicago". The recommendation model is insensitive to context as it generates the same results w.r.t. different contexts.

*Corresponding author.
[1]https://www.baidu.com/
[2]We translate the example from Chinese to English for the sake of understanding.

text queries, Fernández-Tobías and Blanco [2016] proposed a set of memory-based methods that exploit user behaviors in search logs to recommend related entities for a user's full search session. However, these methods purely rely on the users' past behaviors that have been observed in search logs. Therefore, they inevitably suffer from data sparsity and cold start problems, especially for less popular queries and newly introduced entities due to no user behaviors being observed for them.

In this paper, we study the problem of context-aware entity recommendation, and investigate how to use the preceding queries as contexts to improve the recommendation quality. Our approach is based on neural networks, which maps both queries and candidate entities to be recommended into vector space. On top of that, we use attention mechanism to selectively use the in-session preceding context queries to address the ambiguous problem. Furthermore, we improve the model by using a multi-task learning framework, in order to take advantage of the large amounts of cross-task data. Specifically, our multi-task DNN is trained jointly on the tasks of context-aware ranking using click-through data and entity recommendation using entity click logs.

We evaluate our approach using large-scale, real-world search logs of a widely used commercial Web search engine. Experimental results show that incorporating in-session preceding queries significantly improves the performance of entity recommendation. Moreover, the performance is further improved through multi-task learning.

2 Approach

In this section, we formalize the problem, and then detail our approach.

2.1 Problem Definition

Context-Aware Entity Recommendation

In Web search, given a query q_t, the task of entity recommendation [Yu *et al.*, 2014; Huang *et al.*, 2018] is defined as finding a ranked list of entities $E_t = \{e_1, e_2, \cdots, e_h\}$ related to q_t. Traditional recommendation approaches are typically insensitive to the contexts since they only use the current query q_t for generating related entities. In this work, we study context-aware entity recommendation by taking into account the context information. Specifically, given a query q_t, its context C_t, and a set of related entities E_t, our task is to rank the entities in E_t based on the signals derived from both q_t and C_t. In this paper, we assume that the set of entities E_t of q_t is given. In our experiments, E_t is generated by the entity recommendation approach currently employed in a commercial Web search engine (denoted by *Production*).

A *search session* is a period of time which consists of "a sequence of interactions" for the same information need [Shen *et al.*, 2005]. In a search session, a user may interact with the search engine several times. During the interactions, the user would modify her query to achieve desired results for related information needs. Therefore, for the current query q_t (except for the first query in a session, i.e., $t \neq 0$), there is a query history $C_t = q_1, q_2, \cdots, q_{t-1}$ associated with it, which consists of a sequence of preceding queries issued by the user within the same session. In this paper, we consider C_t as the context of the query q_t. Such context information is directly related to the current information need of the user, and is expected to be useful for improving entity recommendation relevance of the current query.

The proportion of sessions in which more than one query was issued is usually not small. Bar-Yossef and Kraus [2011] found that 49% of the queries were preceded by one or more queries in the same session. Xiang *et al.* [2010] reported that about 50% of sessions contain more than one query. We also empirically study a large-scale search log of a commercial Web search engine, which contains 0.42 billion search sessions. Statistics show that 52.61% of the sessions have two or more queries, indicating the substantial potential of using in-session preceding queries as context to improve recommendation quality.

Multi-Task Learning

We also study the problem in a multi-task learning framework, where we consider *context-aware ranking* as an auxiliary task. Context-aware ranking [Shen *et al.*, 2005; Xiang *et al.*, 2010] is defined as ranking the set of documents $D_t = \{d_1, d_2, \cdots, d_w\}$ retrieved for a given query q_t based on the signals derived from both q_t and its context C_t (e.g., preceding queries and/or click-through data) in the same session. The key intuition for using multi-task learning is threefold. First, the two tasks are closely related in Web search and the representations of input queries and contexts can be naturally shared across them. Second, the amount of search logs of context-aware ranking is much larger than that of context-aware entity recommendation. Therefore, it is reasonable to improve entity recommendation by leveraging the abundant search logs of context-aware ranking task in a multi-task learning framework. Third, the clicked documents are helpful in understanding users' search intents behind a query under variant contexts, which can be beneficial to entity recommendation in a multi-task learning framework. For queries with ambiguous or underspecified intents, search results returned by the major commercial Web search engines are often highly diversified [Radlinski and Dumais, 2006; Zhu *et al.*, 2014; Hu *et al.*, 2015] to cover multi-faceted information needs of users. By contrast, the entity recommendation results currently returned for such queries are generally less diverse than the search results, and cannot cover as many intents as possible behind these queries. Therefore, compared with entity recommendation results, it is easier for users to find the desired search results that fulfill their information needs for ambiguous queries. Take the ambiguous query "Chicago" as an example, which may have multiple possible search intents such as city, film, musical, university, travel, and band. Figure 2 shows several clicked results for this query under variant contexts. It shows that the users succeed in finding satisfied search results for this query under both contexts, but fail to find desired entity recommendation results for the less frequent or rare meanings (film or musical) of "Chicago" under the context "Dreamgirls". Intuitively, the clicked documents can be used to infer precise search intents behind a query under a wide variety of contexts, which in turn could help to improve the entity recommendation task.

Query: Chicago	
Context 1: Los Angeles ⇒ Seattle	
Clicked Documents:	Chicago (city)_Baidu Baike Chicago Travel Guide_Baidu Tourism
Clicked Entities:	New York City, Boston
Context 2: Dreamgirls	
Clicked Documents:	Chicago (musical)_Baidu Baike Chicago (Douban)_Douban Movie
Clicked Entities:	N/A

Figure 2: Example of clicked results for the query "Chicago" under variant contexts.

2.2 Basic Entity Recommendation Model

We map both queries and the candidate entities to be recommended to the same vector space, and use vector-based similarity to measure the semantic relevance between a query and an entity. We use a bidirectional LSTM (BiLSTM) [Hochreiter and Schmidhuber, 1997] to encode the query. First, given a query $q = [w_1, w_2, \cdots, w_n]$, the words in q are transformed into vector representations via word embedding matrix, which capture semantic information of words. Then, a forward LSTM and a backward LSTM are employed to map q to a sequence of hidden states $[\overrightarrow{h}_1, \overrightarrow{h}_2, \cdots, \overrightarrow{h}_n]$ and $[\overleftarrow{h}_n, \overleftarrow{h}_{n-1}, \cdots, \overleftarrow{h}_1]$, respectively. Finally, the hidden states $\overrightarrow{h}_n$ and $\overleftarrow{h}_n$ are concatenated as the encoding vector of q: $h_n = [\overrightarrow{h}_n; \overleftarrow{h}_n]$. In this way, we can obtain the representation of q_t, which we denote by v_q. We also embed each entity to the vector space. We denote v_e as the semantic representation of entity e. In this work, we regard an entity as an unique item, which is represented as a continuous vector in the entity embedding matrix. The entity representation could be further improved through taking into account the entity descriptions as suggested in [Xie *et al.*, 2016], which we leave as a future work.

The similarity between a query q and an entity e is computed by cosine similarity:

$$f(q, e) = \cos(v_q, v_e) = \frac{{v_q}^T v_e}{\|v_q\|\|v_e\|}. \quad (1)$$

The parameters of this model could be learned using pairwise learning to rank paradigm [Burges *et al.*, 2005] and stochastic gradient descent. Given a training set $\mathcal{T}_q$, the learning objective is to learn a scoring function $f(q, e)$ that minimizes the negative log likelihood of the clicked entities:

$$-\log \prod_{(q,e^+)\in\mathcal{T}_q} \mathrm{P}(e^+|q), \quad (2)$$

where e^+ is the clicked entity, and the probability of e^+ is computed by:

$$\mathrm{P}(e^+|q) = \frac{\exp(\gamma f(q, e^+))}{\sum_{e\in E} \exp(\gamma f(q, e))}, \quad (3)$$

where E is the set of related entities of q, and γ is a tuning factor determined on held-out data.

2.3 Improved with Contexts from Search Log

In this subsection, we describe our strategy that incorporates the in-session preceding queries as contexts to improve entity recommendation. Our intuition is to get a context-aware query representation that is enhanced by the context. To obtain the representation v_c of context c, we first encode each query in c using the above-mentioned BiLSTM, and get a sequence of encoded queries $v_c^t = [v_{q_1}, v_{q_2}, \cdots, v_{q_{t-1}}]$. Then, we use an attention-based weighted average [Yang *et al.*, 2016] to generate a fixed length vector v_c from v_c^t, which is computed by:

$$v_c = \sum_{i=1}^{t-1} \alpha_i v_{q_i}, \quad (4)$$

where the attention weight α_i is computed by:

$$\alpha_i = \frac{\exp(a_i)}{\sum_{j=1}^{t-1} \exp(a_j)}, \quad (5)$$

$$a_i = v_a^T v_{q_i}, \quad (6)$$

where the latent vector v_a is a trainable parameter and learned during training.

We get a context-aware query representation v_s from the concatenation of v_c and v_q by a fully connected layer. Then, the similarity between q_t, c and e is calculated by:

$$\mathrm{P}(e|c, q_t) = \cos(v_e, v_m) = \frac{{v_e}^T v_m}{\|v_e\|\|v_m\|}, \quad (7)$$

where v_m is the task-specific representation of q_t and c, which is computed from v_s by a fully connected layer.

2.4 Improved with Multi-Task Learning

Multi-task learning is an approach of training multiple tasks in parallel while using a shared representation for knowledge transfer [Caruana, 1997], which has been shown to improve generalization by exploiting the relatedness across tasks. In this paper, we investigate the possibility of leveraging training data collected for the context-aware ranking task to improve the main task of context-aware entity recommendation.

The objective of context-aware ranking is to measure the relevance between a document d and a query q_t with its context c. We use bidirectional LSTM to model both queries and documents. To speed up training with large-scale data, we use document titles instead of entire documents following [Gao *et al.*, 2010] in our experiments. The relevance between q_t, c and d is calculated by:

$$\mathrm{P}(d|c, q_t) = \cos(v_d, v_r) = \frac{{v_d}^T v_r}{\|v_d\|\|v_r\|}, \quad (8)$$

where v_d is the representation of document d, and v_r is the task-specific representation of query q_t and context c, which is computed by a fully connected layer from the shared representation v_s.

Figure 3 shows the architecture of the proposed multi-task DNN model, in which the layers for learning query and context representations are shared across the two tasks (as illustrated in the left part) while other layers are task-specific (as illustrated in the right part). The shared representations are

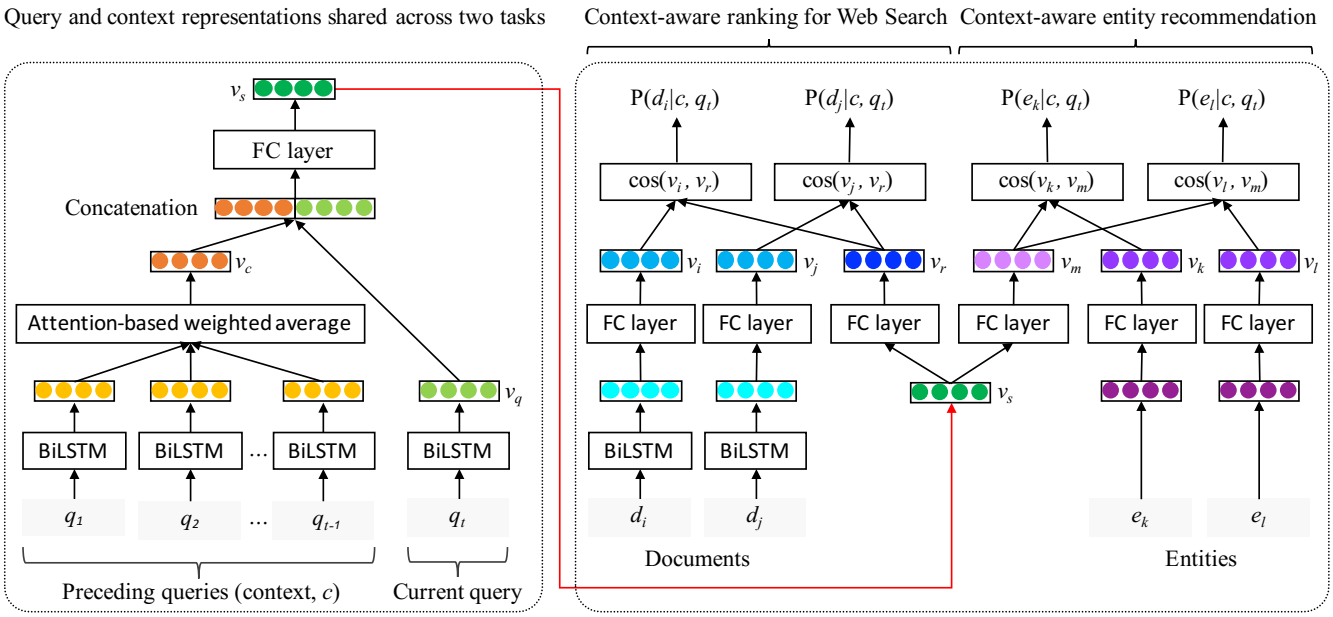

Figure 3: Architecture of the multi-task DNN for representation learning. A bidirectional LSTM (BiLSTM) with attention mechanism is used to learn representations for queries shared across two tasks, while other representations are task-specific. The learned representation v_s which summarizes all preceding queries and current query is used as input for both tasks. FC layer refers to fully connected layer.

trained by the multi-task objective to capture the essential characteristics of contexts and queries, whereas the representations of documents and entities are learned by optimizing the task-specific objectives. In the right part, given a query and its context, the conditional probabilities of an entity and a document are estimated by Equations 7 and 8, respectively.

The parameters of the proposed model are estimated using stochastic gradient descent as described in Algorithm 1. We employ pairwise learning to rank paradigm for both tasks, and the loss functions are the same as Equation 2. We use 2-layer BiLSTM with 128 hidden units. The dimensions of word embeddings, query embeddings, document embeddings, and entity embeddings are set to 256. The mini-batch size is set to 512. The learning rate is initially set to 0.1, which is decayed by a factor of 0.9 after every 10 epochs.

Algorithm 1 Training the multi-task DNN model

1: Initialize model Θ randomly
2: **for** $iteration$ in $1 \cdots I$ **do**
3: Randomly select a task T (context-aware ranking or entity recommendation)
4: Select a random training example for task T
5: Compute loss for task T
6: Compute gradient $\nabla(\Theta)$
7: Update Θ by taking a gradient step with $\nabla(\Theta)$
8: **end for**

2.5 Improved Entity Recommendation Models

After training the multi-task DNN model, we can use it to compute a similarity score between q_t, C_t and each entity $e_c \in E_t$. We investigate two approaches of using the similarity score to build the final context-aware entity recommendation model, either as an individual ranking model or as a feature in a baseline learning to rank framework.

As an Individual Ranking Model

The entities in E_t can be ranked only by comparing the similarity score between each entity $e_c \in E_t$ and q_t with or without C_t. We use the following three models to rank the entities.

- **ER** This model only considers the current query in generating entity recommendations, which is a context-insensitive model as described in subsection 2.2.
- **ER-C** This model uses the in-session preceding queries as contexts to improve entity recommendation, which is a context-aware model as described in subsection 2.3.
- **ER-C-MT** This is the multi-task DNN model that utilizes context information from the context-aware ranking task to improve entity recommendation, as described in subsection 2.4.

As a Feature in a Learning to Rank Framework

The similarity score computed by the above-mentioned models can also be used as a feature in a learning to rank framework. To study the effect of introducing the contextual feature, we employ a set of non-contextual features to train a context-insensitive entity recommendation model denoted by **LTR** and use it as a baseline. This baseline is a competitive model comprising features that have been shown to be effective and strong signals for entity recommendation [Bi *et al.*, 2015]. We omit the description of these features due to space constraints, please refer to [Bi *et al.*, 2015] for more details.

The following three learning to rank models with different similarity features are implemented for comparison.

- **LTR-ER** This model is trained with all LTR features and the similarity feature computed by ER.
- **LTR-ER-C** This model is trained with all LTR features and the similarity feature computed by ER-C.
- **LTR-ER-C-MT** This model is trained with all LTR features and the similarity feature computed by ER-C-MT.

In our experiments, we use stochastic Gradient Boosted Decision Tree (GBDT) [Friedman, 2000] as the learning to rank framework. The parameters of GBDT are tuned using separate training set and validation set.

3 Experiments

3.1 Data Sets and Evaluation Metrics

We evaluate our methods using large-scale, real-world data sets collected from a commercial Web search engine.

First, we collect the training data for context-aware ranking task by extracting the search sessions[3] which contain more than one query and impose the following constraints. Given a session that consists of t ($t > 1$) queries $q_1, q_2, \cdots, q_t$, a training example (C_i, q_i, D_i) is generated such that a document d_i^+ was clicked on for a query q_i ($i > 1$ and $i \leq t$), where C_i is the query history before q_i, i.e., $q_1, \cdots, q_{i-1}$, and D_i is a list of documents that includes d_i^+ (positive example) and K randomly-sampled non-clicked documents $\{d_k^-\}_{k=1,\cdots,K}$ (negative examples).[4] This results in the training data $\mathcal{T}_r = \{(C_i, q_i, D_i)\}$ consisting of 26,426,495 examples. $\mathcal{T}_r$ was randomly split into training set $\mathcal{T}_r^l$ (80%), validation set $\mathcal{T}_r^v$ (10%), and test set $\mathcal{T}_r^t$ (10%).

Second, we use the same method and search sessions to extract training data for learning representations of queries and entities for entity recommendation task. This results in the training data $\mathcal{T}_e = \{(C_j, q_j, E_j)\}$ consisting of 8,821,550 examples, where E_j is a list of entities that includes e_j^+ and L randomly-sampled non-clicked entities $\{d_l^-\}_{l=1,\cdots,L}$.[5] $\mathcal{T}_e$ was randomly split into training set $\mathcal{T}_e^l$ (80%) and validation set $\mathcal{T}_e^v$ (20%). $\mathcal{T}_e^l$ is used to train ER-C. The data set obtained by removing contexts from $\mathcal{T}_e^l$ is used to train ER. $\mathcal{T}_e^l$ and $\mathcal{T}_r^l$ are used to train the multi-task DNN model ER-C-MT.

Existing entity recommendation systems tend to recommend entities for ambiguous queries with regard to the frequently asked meaning of them as we discussed in Section 1. Therefore, there may be imbalance in the data set $\mathcal{T}_e$ for frequent and rare meanings of ambiguous queries, especially for the rare meanings of such queries that have no entity clicks. To conduct better evaluation, we need a data collection approach which can alleviate this problem to some extent. Although manually labeling the relevance of each recommended entity w.r.t. both a given query and its preceding queries is a straightforward way, there are two concerns. First, it is expensive and limited in quantity. Second, the relevance judged by annotators may not be consistent with that inferred from the observed behaviors of real search users. Therefore, we decide to build the test set and evaluate the performance of our methods by using real click data.

We collect this data by using online sampling method. Specifically, given a query q_s which has at least one preceding query as its context C_s, we randomly select a list of entities from the set of candidate entities[6] of q_s as recommendations to a user when she searches for q_s. During the procedure, whether the user clicks a recommended entity after she searches for q_s under the context C_s is logged. We sample a small portion of queries and search users for testing and run the procedure during a 15-day period. In this way, we can obtain a data set consisting of (C_s, q_s, E_s), where $E_s = \{(e_o, c_o)\}$, and c_o is the aggregated clicks of an entity e_o clicked on query q_s under the context C_s. To conduct fair evaluation, a sample (C_s, q_s, E_s) would be filtered if the aggregated clicks in E_s are all the same, since it is useless for evaluating a ranking model. This results in a data set $\mathcal{T} = \{(C_s, q_s, E_s)\}$ consisting of 8,402,881 examples. $\mathcal{T}$ was randomly split into training set $\mathcal{T}_l$ (80%), validation set $\mathcal{T}_v$ (10%), and test set $\mathcal{T}_t$ (10%). $\mathcal{T}_l$ and $\mathcal{T}_v$ are used to train and tune all learning to rank models, while $\mathcal{T}_t$ is used to evaluate all entity recommendation models.

We employ NDCG [Järvelin and Kekäläinen, 2002] to evaluate our methods, which is a commonly used metric for evaluating ranked results in information retrieval.

3.2 Baseline Method

To evaluate our proposed model, we use the memory-based approach proposed by [Fernández-Tobías and Blanco, 2016] as baseline method (denoted by **MBR**) for comparison.[7] This method is based on nearest neighbors collaborative filtering [Sarwar *et al.*, 2001; Linden *et al.*, 2003] and purely relies on user behaviors in search logs to recommend related entities for a user's full search session. The probability of an entity e being relevant for a session s is estimated by:

$$\mathrm{P}(e|s) = \sum_{\bar{e} \in E(s)} \mathrm{P}(e|\bar{e})\mathrm{P}(\bar{e}|s), \tag{9}$$

where $E(s)$ is the set of clicked entities in session s. $\mathrm{P}(e|\bar{e})$ captures the similarity between a pair of entities and is estimated by co-occurrence of entities in search sessions using Jaccard's coefficient. $\mathrm{P}(\bar{e}|s)$ estimates how relevant the clicked entity $\bar{e}$ is in session s and is computed by:

$$\mathrm{P}(\bar{e}|s) = \sum_q \mathrm{P}(\bar{e}, q|s) = \sum_q \mathrm{P}(\bar{e}|q, s)\mathrm{P}(q|s), \tag{10}$$

where $\mathrm{P}(\bar{e}|q, s)$ is the importance of $\bar{e}$ for query q, and $\mathrm{P}(q|s)$ is the query likelihood of q in session s.

[3] We completely anonymized the user data, and then segmented each user's stream into sessions using a commonly used rule [White *et al.*, 2007; Jansen *et al.*, 2007], i.e., a boundary between two sessions was set by user inactivity (either no query or no click) for more than 30 minutes. In our experiments, the search sessions were sampled from a 3-month period of a commercial Web search engine.

[4] We set K=3 in our experiments.

[5] We set L=3 in our experiments.

[6] Top-ranked 100 entities generated by *Production* are used here.

[7] The baseline model is trained on $\mathcal{T}_e^l$ and evaluated on $\mathcal{T}_t$.

	NDCG@1	NDCG@5	NDCG@10
MBR	0.0194	0.0444	0.0641
ER	0.0203	0.0454	0.0663
ER-C	0.0206	0.0455	0.0675
ER-C-MT	0.0216	0.0504	0.0710
LTR	0.1219	0.2103	0.2502
LTR-ER	0.1332	0.2261	0.2665
LTR-ER-C	0.1386	0.2324	0.2728
LTR-ER-C-MT	**0.1461**$^{\blacktriangle}$	**0.2438**$^{\blacktriangle}$	**0.2834**$^{\blacktriangle}$

Table 1: The evaluation results of different models.

	Short	Medium	Long
MBR	0.0538	0.0618	0.0690
ER	0.0688	0.0689	0.0641
ER-C	0.0704	0.0705	0.0648
ER-C-MT	0.0752	0.0741	0.0679
LTR	0.2553	0.2648	0.2409
LTR-ER	0.2715	0.2824	0.2566
LTR-ER-C	0.2776	0.2898	0.2624
LTR-ER-C-MT	**0.2883**$^{\blacktriangle}$	**0.3016**$^{\blacktriangle}$	**0.2722**$^{\blacktriangle}$

Table 2: Results of each model on contexts with different lengths.

3.3 Results and Analysis

We report empirical results and analysis in this subsection. We perform 10-fold cross validation and test for statistical significance using a paired two-tailed t-test. In each table, we depict statistical significance of the best result over all other results in the same column with $p < 0.01$ by $^{\blacktriangle}$. Boldface indicates the highest score w.r.t. each metric.

We first evaluate whether our model can improve entity recommendation performance. From the evaluation results in Table 1, we observe that: 1) ER, ER-C, and ER-C-MT all perform better than the baseline model MBR. This demonstrates the superior performance of neural network based models; 2) both ER-C and ER-C-MT outperform ER, which indicates that preceding queries are useful for improving entity recommendation relevance of the current query; 3) both LTR-ER-C and LTR-ER-C-MT significantly outperform LTR and LTR-ER, which demonstrates that context information can significantly help to improve the performance of entity recommendation; and 4) ER-C-MT significantly outperforms ER-C, and LTR-ER-C-MT significantly outperforms LTR-ER-C, which shows the effectiveness of the multi-task objective (including both context-aware ranking and entity recommendation) over the single-task objective (only entity recommendation).

Then, we investigate the effect of context length on the performance of our model. Following [Sordoni *et al.*, 2015; Dehghani *et al.*, 2017], we separate the test set $\mathcal{T}_t$ into three categories: 1) contexts with 1 query (short) 17.81% of $\mathcal{T}_t$; 2) contexts with 2-3 queries (medium) 28.31% of $\mathcal{T}_t$; and 3) contexts with >3 queries (long) 53.88% of $\mathcal{T}_t$. Table 2 shows the performance of each model in terms of NDCG@10 on contexts with different lengths.[8] From the results, we observe that: 1) LTR-ER-C-MT robustly performs best across the test sets with variant context lengths, which further confirms the effectiveness of our model; 2) ER-C-MT significantly outperforms ER-C, and LTR-ER-C-MT significantly outperforms LTR-ER-C. This shows the effectiveness of the proposed multi-task DNN on different context lengths; and 3) it seems that all our context-aware models (ER-C, ER-C-MT, LTR-ER-C, and LTR-ER-C-MT) achieve better results on short and medium contexts than that on long contexts. A possible reason is that information needs are topically broad or there exists topic drifts within long sessions, making it difficult to model the relevance between queries in such sessions. However, MBR achieves the best result on long contexts among all the contexts, and it also obtains the best performance on long contexts in comparison with ER, ER-C, and ER-C-MT. The main reason is that the method used by MBR to model topic drift within a session is highly time-sensitive, which assumes that the queries issued earlier are likely to be less representative of the current user task. As a result, queries that are issued too far away from the current query will be considered as non-relevant and discarded.

To better illustrate the effectiveness of context information, we show several examples of LTR-ER-C-MT (context-aware model) and LTR (context-insensitive model) in Figure 4. We can see that the entity recommendations generated by LTR-ER-C-MT are better than those generated by LTR, because they are relevant to both the query and the context.

Finally, we also empirically investigate whether the model trained with the multi-task objective can improve the performance of context-aware ranking. To this end, we trained a single-task DNN model CR-ST on the same training set $\mathcal{T}_r^l$ using the single-task objective (only context-aware ranking). We evaluated ER-C-MT and CR-ST on the same test set $\mathcal{T}_r^t$. The results in Table 3 show that ER-C-MT significantly

Query and Context
Query: A Song of Ice and Fire
Context: Maisie Williams ⇒ Rose Leslie
Entity Recommendations
LTR: Westworld, Game of Thrones, House of Card, Nip/Tuck, Frozen
LTR-ER-C-MT: Isaac Hempstead-Wright, Carice van Houten, Iwan Rheon, Liam Cunningham, Peter Dinklage

Query and Context
Query: Florence
Context: Soccer Players ⇒ Roberto Baggio
Entity Recommendations
LTR: Vatican City, Pompeii, Rome, Metropolitan City of Florence, San Gimignano
LTR-ER-C-MT: A.C. Milan, A.S. Roma, Inter Milan, A.C. ChievoVerona, Real Madrid C.F.

Figure 4: The entity recommendations generated by different methods on the test queries with contexts taken from search logs.

[8] We omit NDCG@1 and NDCG@5 due to space constraints. Statistics show that LTR-ER-C-MT also significantly outperforms all other models in terms of both NDCG@1 and NDCG@5.

	NDCG@1	NDCG@5	NDCG@10
CR-ST	0.2735	0.4849	0.5860
ER-C-MT	**0.2742**▲	**0.4856**▲	**0.5867**▲

Table 3: Results of different context-aware ranking models.

outperforms CR-ST. This indicates that the task of context-aware ranking can also benefit from the regularization effect of multi-task learning, which helps to reduce overfitting of the learned representations to a specific task [Liu *et al.*, 2015].

4 Related Work

Previous work that is the closest to our task is the task of entity recommendation, e.g., [Blanco *et al.*, 2013; Yu *et al.*, 2014; Bi *et al.*, 2015; Huang *et al.*, 2018]. However, none of them are context-aware in that they do not take into account the in-session preceding queries as context. By contrast, the task of context-aware entity recommendation requires to consider and understand the context information, and use it effectively in entity recommendations. Huang *et al.* [2016; 2017] proposed to enhance the understandability of entity recommendations by captioning the results. However, they were also insensitive to context. To better identify a user's search needs and provide more relevant results, entity recommendation should be context-aware and account for the preceding queries issued by the user within a search session. Recently, to tackle the above challenges, Fernández-Tobías and Blanco [2016] proposed a number of memory-based methods that exploit user behaviors in search logs to recommend related entities for a user's full search session. However, these methods highly rely on the users' past behaviors that have been observed in search logs. Therefore, they inevitably suffer from data sparsity and cold start problems.

The context which consists of in-session preceding queries has proven to be effective in helping to understand a user's current information need, and plays an important role in improving the performance of several search applications, such as query suggestion [Cao *et al.*, 2008; Mitra, 2015; Sordoni *et al.*, 2015; Dehghani *et al.*, 2017] and context-aware ranking for Web search [Shen *et al.*, 2005; Xiang *et al.*, 2010; Li *et al.*, 2014]. However, these methods are based on single-task supervised learning with sufficient training data that are sensitive to context. By contrast, the training data collected from entity click logs of a context-insensitive recommendation system are typically insensitive to the contexts. To address this problem, we consider context-aware ranking as the auxiliary task, and further develop a multi-task DNN that leverages the context-specific information contained in the training data of this task to improve upon the main task of entity recommendation.

The use of multi-task learning [Caruana, 1997] with DNNs in our task is inspired by the recent remarkable success of applying it in various natural language processing tasks. For example, Collobert *et al.* [2011] proposed to learn representations shared across multiple tasks of part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. Bordes *et al.* [2012] proposed to jointly learn representations of words and entities via multi-task training on multiple data sources for relation extraction. Dong *et al.* [2015] proposed an NMT model under the multi-task learning framework to address the problem of translating one source language into multiple target languages. Liu *et al.* [2015] proposed a multi-task DNN model to combine two tasks of query classification and ranking for Web search, and achieved improvement on both tasks. While conceptually similar, our model is novel in that it successfully combines the tasks of context-aware ranking and entity recommendation, which facilitates context acquisition and context modeling for the latter by leveraging context information contained in the former.

5 Conclusion

In this paper, we study the problem of context modeling for improving entity recommendation. To this end, we develop a multi-task DNN that learns representations across multiple tasks by leveraging large amounts of cross-task data. We evaluate our approach using large-scale, real-world search logs of a widely used commercial Web search engine. The experiments demonstrate that context information can significantly improve the performance of entity recommendation.

Generally two categories of context information can be derived from search logs. One is short-term context such as preceding queries or clicked documents in a session. Another is long-term context such as search history or click-through data across all sessions. As future work, we plan to investigate whether long-term context or other short-term context (e.g., preceding clicked documents before a user's current search in a session) could help to improve entity recommendation.

Acknowledgments

This research is supported by the National Basic Research Program of China (973 program No. 2014CB340505). We would like to thank the anonymous reviewers for their insightful comments.

References

[Bar-Yossef and Kraus, 2011] Ziv Bar-Yossef and Naama Kraus. Context-sensitive query auto-completion. In *WWW*, pages 107–116, 2011.

[Bi *et al.*, 2015] Bin Bi, Hao Ma, Bo-June (Paul) Hsu, Wei Chu, Kuansan Wang, and Junghoo Cho. Learning to recommend related entities to search users. In *WSDM*, pages 139–148, 2015.

[Blanco *et al.*, 2013] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. Entity recommendations in web search. In *ISWC*, pages 33–48, 2013.

[Bordes *et al.*, 2012] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *AISTATS*, pages 127–135, 2012.

[Burges *et al.*, 2005] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005.

[Cao *et al.*, 2008] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *SIGKDD*, pages 875–883, 2008.

[Caruana, 1997] Rich Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, 1997.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

[Dehghani *et al.*, 2017] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. Learning to attend, copy, and generate for session-based query suggestion. In *CIKM*, pages 1747–1756, 2017.

[Dong *et al.*, 2015] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *ACL*, pages 1723–1732, 2015.

[Fernández-Tobías and Blanco, 2016] Ignacio Fernández-Tobías and Roi Blanco. Memory-based recommendations of entities for web search users. In *CIKM*, pages 35–44, 2016.

[Friedman, 2000] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[Gao *et al.*, 2010] Jianfeng Gao, Xiaodong He, and Jian-Yun Nie. Clickthrough-based translation models for web search: from word models to phrase models. In *CIKM*, pages 1139–1148, 2010.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hu *et al.*, 2015] Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. Search result diversification based on hierarchical intents. In *CIKM*, pages 63–72, 2015.

[Huang *et al.*, 2016] Jizhou Huang, Shiqi Zhao, Shiqiang Ding, Haiyang Wu, Mingming Sun, and Haifeng Wang. Generating recommendation evidence using translation model. In *IJCAI*, pages 2810–2816, 2016.

[Huang *et al.*, 2017] Jizhou Huang, Wei Zhang, Shiqi Zhao, Shiqiang Ding, and Haifeng Wang. Learning to explain entity relationships by pairwise ranking with convolutional neural networks. In *IJCAI*, pages 4018–4025, 2017.

[Huang *et al.*, 2018] Jizhou Huang, Shiqiang Ding, Haifeng Wang, and Ting Liu. Learning to recommend related entities with serendipity for web search users. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 17(3):25:1–25:22, April 2018.

[Jansen *et al.*, 2007] Bernard J. Jansen, Amanda Spink, Chris Blakely, and Sherry Koshman. Defining a session on web search engines. *J. Am. Soc. Inf. Sci. Technol.*, 58(6):862–871, 2007.

[Järvelin and Kekäläinen, 2002] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[Li *et al.*, 2014] Xiujun Li, Chenlei Guo, Wei Chu, Ye-Yi Wang, and Jude Shavlik. Deep learning powered in-session contextual ranking using clickthrough data. In *NIPS*, 2014.

[Linden *et al.*, 2003] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

[Liu *et al.*, 2015] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *NAACL*, pages 912–921, 2015.

[Mitra, 2015] Bhaskar Mitra. Exploring session context using distributed representations of queries and reformulations. In *SIGIR*, pages 3–12, 2015.

[Radlinski and Dumais, 2006] Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *SIGIR*, pages 691–692, 2006.

[Sarwar *et al.*, 2001] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[Shen *et al.*, 2005] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR*, pages 43–50, 2005.

[Sordoni *et al.*, 2015] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *CIKM*, pages 553–562, 2015.

[White *et al.*, 2007] Ryen W. White, Mikhail Bilenko, and Silviu Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *SIGIR*, pages 159–166, 2007.

[Xiang *et al.*, 2010] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware ranking in web search. In *SIGIR*, pages 451–458, 2010.

[Xie *et al.*, 2016] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *AAAI*, pages 2659–2665, 2016.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *NAACL*, pages 1480–1489, 2016.

[Yu *et al.*, 2014] Xiao Yu, Hao Ma, Bo-june Paul Hsu, and Jiawei Han. On building entity recommender systems using user click log and freebase knowledge. In *WSDM*, pages 263–272, 2014.

[Zhu *et al.*, 2014] Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. Learning for search result diversification. In *SIGIR*, pages 293–302, 2014.